

Text Generation

TEK5040/TEK9040

Eilif Solberg

In this weeks assignment we will look at the [tutorial on text generation with RNN](#). Open the link in your browser and perform the following steps.

1. Choose 'Run in Google Colab'. Go through the tutorial as is and make sure you understand all the steps. We don't generally use `model.compile` and `model.fit` in this course as they may seem a bit "magical". As an exercise you may try to implement the optimization without using these functions. At any rate you may see such an implementation in the 'advanced' section at the end. This custom training loop really does exactly the same steps as is performed by `model.compile` and `model.fit`, there is really no magic to these functions!
2. Change the GRU layer to an LSTM layer. Do you notice any difference in the loss?
3. Create a "deep" RNN by stacking a second LSTM layer on top of the previous layer in `build_model`. You may use either `model.fit` or the custom training loop to train your model.
4. In class we implemented a verion of the layer that is called `LSTMCell` in TensorFlow. In TensorFlow, what is the difference between the `LSTM` and `LSTMCell` layers (or in general between `RNN` and `RNNCell` layers)? *After* you have considered this question you may want to check the [Keras RNN guide](#) which explains this relationship and more.
5. OPTIONAL. Create a deep RNN by implementing a custom deep RNN cell (the other way of making RNNs deep as discussed in the lecture), e.g. as a multilayer perceptron, and wrap it with `keras.layers.RNN` (https://www.tensorflow.org/api_docs/python/tf/keras/layers/RNN).

You are of course free to play around and try out other things!